How to Index Virtual Fields

Releases
505_12.2
500_12.3
505_12.4

Last update: 24 September, 2000

Ex Libris

Table of Contents

Introduction

In ALEPH, the concept of expanded or virtual fields is widespread. These fields are used for indexing and for display throughout the system. There are essentially two categories of expand fields: those that rely on expand programs alone and those that can be configured using the four expand tables. Frequently, the program-driven expand fields take data from one database and display/index it in another database. Examples of this are the LOC and PSTS fields, which take data from the administrative and holdings databases and expand them into the bibliographic database. Table-driven expand fields, on the other hand, manipulate data that appears in a single record and a single database. This document deals with the table-driven expand programs only. There are currently four tables for configuring expand fields. They are: tab_expand_extract, tab_expand_join, tab_expand_join_simple, and tab_expand_split.

Expand_extract

Tab_expand_extract supports the extraction of individual instances of a sub-field for display and indexing. This table should be used to index a single instance of a repeatable sub-field. If the sub-field is not repeatable or if you want to index all occurrences of the sub-field as one entry, there is no need to use this table. However, if the sub-field is repeatable and you want individual index entries for each occurrence, expand_extract should be used.

Example One:

Field: 650 0 L $$aMedical care$$zManitoba$$zWinnipeg$$xHistory.

Entry without expand_extract: $$zManitoba$$zWinnipeg

Entry with expand_extract: $$zManitoba
$$zWinnipeg

Expand_join

Tab_expand_join supports the concatenation of multiple fields for display and indexing. This table can be used to concatenate actual fields (see the following Example One) as well as virtual fields (Example Two).

Example One:

Fields: 1001 L $$aGuinther, John.
24514 L $$aThe malpractitioners /$$cby John Guinther. --

Joined fields: AT100 L $$aGuinther, John.$$amalpractitioners /

Example Two:

Fields: LOC0 L $$bXLINC$$cGEN$$hPR3523$$iS8
24510 L $$aSamuel Johnson's literary criticism.$$cEdited by R. D. Stock. –

Joined fields: CNT00 $$aPR3523$$bS8$$cSamuel Johnson's literary criticism.

Expand_join_simple

Like expand_join, expand_join_simple creates a new field by concatenating two or more fields in the bibliographic record. The difference between the two tables is that while the expand_join routine only creates a virtual field when two or more of the fields being concatenated are present, expand_join_simple creates a virtual field when only one of the concatenated fields is present. This makes expand_join_simple especially well suited for display and expand_join well suited for indexing.

Example One:

Fields: 24514 L $$aThe malpractitioners /$$cby John Guinther. --
250 L $$a1st ed. --

Joined fields: TMP11 L $$aThe malpractitioners /$$y1st ed. –

Example Two:

Fields: 24500 L $$aBeneficent euthanasia.$$cEdited by Marvin Kohl. --
No 250

Joined fields: TMP11 L $$aBeneficent euthanasia

Expand_split

Tab_expand_split extracts individual occurences of a sub-field string by creating an entirely new instance of the field when the string of sub-fields repeats itself. A good example of how this program routine works is the imprint field – occasionally there will be more then one imprint, sub-fields a and b of the 260 repeat, the order is $$a $$b $$a $$b.

Example One:

Field: 260 L$$aNew-York,$$bScribner$$aBerlin,$bSpringer-Verlag
Split fields: 260 L$$aNew-York,$$bScribner

- L$$aBerlin,$$bSpringer-Verlag

Indexing expand fields

In order to index an expanded field, you will need to:

- Set up the appropriate expand table.
- Add the expand field to tab11.
- Call the expand program in the appropriate section of tab_expand.
- Re-run the appropriate index.

Setting up the tables:

Follow instructions contained in the table header or in the Database Management Guide for your version. Note that both tab_expand_join and tab_expand_join_simple will strip non-filing characters from the joined field using the appropriate indicators. The indicator values will NOT be transferred to the new field. Thus, five characters, not three, are available for naming the new field.

Example One:

tab_expand_extract
650## z **z650 [field name]**

tab_expand_join
AU100 100## -e46 abcde 240## -h6 fghij **[field name]**

tab_expand_join_simple
TMP11 245## AA a a 250## 01 ac yz **[field name]**

tab_expand_split
260## a

Add the expand field to tab11

Tab11 is the ALEPH configuration table responsible for indexing. Three of the four expand field tables create virtual fields with tags that will not conform to MARC21 conventions. These fields will not be present in tab11 and thus will not be indexed unless you add them to your tab11. The name of the field you created is identified in the expand_table - Col. 3 of tab_expand_extract, col. 1 of tab_expand_join and tab_expand_join_simple. Remember, expand_split creates a new instance of the field identified in the table. The new field will have the same tag as the field from which it is derived. There is no need to add a new field code to tab11.

Add the expand field to tab_sort

Virtual fields, including those you create using the four expand tables, can be used to generate sort keys. Instead of adding the expand field code to tab11 you will need to add it to tab_sort.

Call the expand program in the ACC and/or WORD section of tab_expand

After adding the expand field to tab11, be sure to call the correct expand routine in the appropriate section of tab_expand. For keyword indexes, the routine should be added to the section in tab_expand marked WORD. For headings, it should be added to the section marked ACC. For direct indexes, it should be added to the section marked INDEX, and for sort indexes it should be added to the section marked SORT-DOC.

Example:

```
tab_expand
WORD      expand_doc_fmt_mgu
WORD      expand_doc_join
!
ACC       expand_doc_bib_loc_usm
ACC       expand_doc_bib_ndu
ACC       expand_doc_join
!
INDEX     expand_doc_extract
INDEX     expand_doc_bib_loc_usm
!
SORT-DOC   expand_doc_bib_loc_usm
```

Note:

If you are joining or extracting virtual fields, the order in which the expand routines appear in tab_expand IS signifigant. To join a LOC field with a 245 field in order to create a merged call number-title index, first call expand_doc_bib_loc_usm and then call expand_doc_join.

Re-run the appropriate index

After making changes to your library's indexing tables, you must re-run the batch processes responsible for rebuilding the indexes affected by your changes. The mode in which you run these processes will depend on the type of changes you have made. If new indexes have been added, it is sufficient to run the processes in update mode. However, if existing indexes have been changed, you will have to run the processes in rebuild mode.